# Package ‘itsmr’

September 10, 2018

**Type** Package  
**Title** Time Series Analysis Using the Innovations Algorithm  
**Version** 1.9  
**Date** 2018-09-06  
**Author** George Weigt  
**Maintainer** George Weigt <g808391@icloud.com>  
**Description** Provides functions for modeling and forecasting time series data. Forecasting is based on the innovations algorithm. A description of the innovations algorithm can be found in the textbook "Introduction to Time Series and Forecasting" by Peter J. Brockwell and Richard A. Davis. <http://www.springer.com/us/book/9781475777505>.  
**License** FreeBSD  
**LazyLoad** yes  
**NeedsCompilation** no  
**URL** http://eigenmath.org/itsmr-refman.pdf  
**Repository** CRAN  
**Date/Publication** 2018-09-10 18:30:02 UTC

### R topics documented:

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>itsmr-package</td>
<td>2</td>
</tr>
<tr>
<td>aacvf</td>
<td>3</td>
</tr>
<tr>
<td>acvf</td>
<td>4</td>
</tr>
<tr>
<td>airpass</td>
<td>5</td>
</tr>
<tr>
<td>ar.inf</td>
<td>5</td>
</tr>
<tr>
<td>arar</td>
<td>6</td>
</tr>
<tr>
<td>arma</td>
<td>7</td>
</tr>
<tr>
<td>autofit</td>
<td>8</td>
</tr>
<tr>
<td>burg</td>
<td>9</td>
</tr>
<tr>
<td>check</td>
<td>10</td>
</tr>
<tr>
<td>deaths</td>
<td>10</td>
</tr>
<tr>
<td>dowj</td>
<td>11</td>
</tr>
</tbody>
</table>
Description

Provides functions for modeling and forecasting time series data. Forecasting is based on the innovations algorithm. A description of the innovations algorithm can be found in the textbook *Introduction to Time Series and Forecasting* by Peter J. Brockwell and Richard A. Davis.

Details

Package: itsmr
Type: Package
Version: 1.9
Date: 2018-09-06
License: FreeBSD
LazyLoad: yes
URL: http://eigenmath.org/itsmr-refman.pdf
aacvf

Author(s)
George Weigt
Maintainer: George Weigt <g808391@icloud.com>

References

Examples
```r
plotc(wine)

## Define a suitable data model
M = c("log", "season", 12, "trend", 1)

## Obtain residuals and check for stationarity
e = resid(wine, M)
test(e)

## Define a suitable ARMA model
a = arma(e, p=1, q=1)

## Obtain residuals and check for white noise
ee = resid(wine, M, a)
test(ee)

## Forecast future values
forecast(wine, M, a)
```

aacvf

**Autocovariance of ARMA model**

Description
Autocovariance of ARMA model

Usage
```r
aacvf(a, h)
```

Arguments
- **a**: ARMA model
- **h**: Maximum lag
Details

The ARMA model is a list with the following components.

- \( \phi \) Vector of AR coefficients (index number equals coefficient subscript)
- \( \theta \) Vector of MA coefficients (index number equals coefficient subscript)
- \( \sigma_2 \) White noise variance

Value

Returns a vector of length \( h+1 \) to accommodate lag 0 at index 1.

See Also

arma

Examples

```r
a = arma(Sunspots, 2, 0)
aacf(a, 40)
```
airpass  

Number of international airline passengers, 1949 to 1960

Description

Number of international airline passengers, 1949 to 1960

Examples

plotc(airpass)

ar.inf  

Compute AR infinity coefficients

Description

Compute AR infinity coefficients

Usage

ar.inf(a, n = 50)

Arguments

a  ARMA model
n  Order

Details

The ARMA model is a list with the following components.

phi  Vector of AR coefficients (index number equals coefficient subscript)
theta  Vector of MA coefficients (index number equals coefficient subscript)
sigma2  White noise variance

Value

Returns a vector of length n+1 to accomodate coefficient 0 at index 1.

See Also

ma.inf
Examples

a = yw(Sunspots, 2)
ar.ar(a)

arar Forecast using ARAR algorithm

Description

Forecast using ARAR algorithm

Usage

arar(y, h = 10, opt = 2)

Arguments

y Time series data
h Steps ahead
opt Display option (0 silent, 1 tabulate, 2 plot and tabulate)

Value

Returns the following list invisibly.

pred Predicted values
se Standard errors
l Lower bounds (95% confidence interval)
u Upper bounds

See Also

forecast

Examples

arar(airpass)
Estimate ARMA model coefficients using maximum likelihood

arma(x, p = 0, q = 0)

Arguments

- **x**: Time series data
- **p**: AR order
- **q**: MA order

Details

Calls the standard R function `arima` to estimate AR and MA coefficients. The innovations algorithm is used to estimate white noise variance.

Value

Returns an ARMA model consisting of a list with the following components.

- **phi**: Vector of AR coefficients (index number equals coefficient subscript)
- **theta**: Vector of MA coefficients (index number equals coefficient subscript)
- **sigma2**: White noise variance
- **aicc**: Akaike information criterion corrected
- **se.phi**: Standard errors for the AR coefficients
- **se.theta**: Standard errors for the MA coefficients

See Also

- `autofit`, `burg`, `hannan`, `ia`, `yw`

Examples

```r
M = c("diff",1)
e = Resid(dowj,M)
a = arma(e,1,0)
print(a)
```
autofit  

Find the best model from a range of possible ARMA models

Description

Find the best model from a range of possible ARMA models

Usage

autofit(x, p = 0:5, q = 0:5)

Arguments

x  
Time series data (typically residuals from Resid)

p  
Range of AR orders

q  
Range of MA orders

Details

Tries all combinations of p and q and returns the model with the lowest AICC. The arguments p and q should be small ranges as this function can be slow otherwise. The innovations algorithm is used to estimate white noise variance.

Value

Returns an ARMA model consisting of a list with the following components.

phi  
Vector of AR coefficients (index number equals coefficient subscript)

theta  
Vector of MA coefficients (index number equals coefficient subscript)

sigma2  
White noise variance

aicc  
Akaike information criterion corrected

se.phi  
Standard errors for the AR coefficients

se.theta  
Standard errors for the MA coefficients

See Also

arma

Examples

M = c("diff",1)
e = Resid(dowj,M)
a = autofit(e)
print(a)
Estimate AR coefficients using the Burg method

Usage

burg(x, p)

Arguments

x Time series data (typically residuals from `Resid`)

p AR order

Details

The innovations algorithm is used to estimate white noise variance.

Value

Returns an ARMA model consisting of a list with the following components.

- phi Vector of AR coefficients (index number equals coefficient subscript)
- theta 0
- sigma2 White noise variance
- aicc Akaike information criterion corrected
- se.phi Standard errors for the AR coefficients
- se.theta 0

See Also

arma hannan ia yw

Examples

```r
M = c("diff", 1)
e = Resid(dowj, M)
a = burg(e, 1)
print(a)
```
check | **Check for causality and invertibility**

**Description**

Check for causality and invertibility

**Usage**

`check(a)`

**Arguments**

`a` ARMA model

**Details**

The ARMA model is a list with the following components.

- `phi` Vector of AR coefficients (index number equals coefficient subscript)
- `theta` Vector of MA coefficients (index number equals coefficient subscript)
- `sigma2` White noise variance

**Value**

None

**Examples**

```r
a = specify(ar=c(0,0.99))
check(a)
```

---

**Deaths**

**USA accidental deaths, 1973 to 1978**

**Description**

USA accidental deaths, 1973 to 1978

**Examples**

```r
plotc(deaths)
```
**dowj**

*Dow Jones utilities index, August 28 to December 18, 1972*

**Description**

Dow Jones utilities index, August 28 to December 18, 1972

**Examples**

```r
plotc(dowj)
```

---

**forecast**

*Forecast future values*

**Description**

Forecast future values

**Usage**

```r
forecast(x, M, a, h = 10, opt = 2, alpha = 0.05)
```

**Arguments**

- **x**: Time series data
- **M**: Data model
- **a**: ARMA model
- **h**: Steps ahead
- **opt**: Display option (0 silent, 1 tabulate, 2 plot and tabulate)
- **alpha**: Level of significance

**Details**

The data model can be *NULL* for none. Otherwise *M* is a vector of function names and arguments.

Example:

```r
M = c("log","season",12,"trend",1)
```

The above model takes the log of the data, then subtracts a seasonal component of period 12, then subtracts a linear trend component.

These are the available functions:

- **diff**: Difference the data. Has a single argument, the lag.
- **hr**: Subtract harmonic components. Has one or more arguments, each specifying the number of observations per harmonic.
- **log**: Take the log of the data, has no arguments.
- **season**: Subtract a seasonal component. Has a single argument, the number of observations per season.
- **trend**: Subtract a trend component. Has a single argument, the order of the trend (1 linear, 2 quadratic, etc.)
At the end of the model there is an implicit subtraction of the mean operation. Hence the resulting time series always has zero mean. All of the functions are inverted before the forecast results are displayed.

Value

Returns the following list invisibly.

- `pred`: Predicted values
- `se`: Standard errors (not returned for data models with log)
- `l`: Lower bounds (95% confidence interval)
- `u`: Upper bounds

See Also

- `arma` 
- `Resid` 
- `test`

Examples

```r
M = c("log", "season", 12, "trend", 1)
e = Resid(wine, M)
a = arma(e, 1, 1)
forecast(wine, M, a)
```

hannan

Estimate ARMA coefficients using the Hannan-Rissanen algorithm

Description

Estimate ARMA coefficients using the Hannan-Rissanen algorithm

Usage

```r
hannan(x, p, q)
```

Arguments

- `x`: Time series data (typically residuals from `Resid`)
- `p`: AR order
- `q`: MA order (q > 0)

Details

The innovations algorithm is used to estimate white noise variance.
hr

Value
Returns an ARMA model consisting of a list with the following components.

- **phi**: Vector of AR coefficients (index number equals coefficient subscript)
- **theta**: Vector of MA coefficients (index number equals coefficient subscript)
- **sigma2**: White noise variance
- **aicc**: Akaike information criterion corrected
- **se.phi**: Standard errors for the AR coefficients
- **se.theta**: Standard errors for the MA coefficients

See Also
arma burg ia yw

Examples

```r
M = c("diff",12)
e = Resid(deaths,M)
a = hannan(e,1,1)
print(a)
```

hr

Estimate harmonic components

Description
Estimate harmonic components

Usage

hr(x, d)

Arguments

- **x**: Time series data
- **d**: Vector of harmonic periods

Value

Returns a vector the same length as x. Subtract from x to obtain residuals.

Examples

```r
y = hr(deaths,c(12,6))
plotc(deaths,y)
```
Estimate MA coefficients using the innovations algorithm

Description

Estimate MA coefficients using the innovations algorithm

Usage

ia(x, q, m = 17)

Arguments

x          Time series data (typically residuals from Resid)
q          MA order
m          Recursion level

Details

Normally m should be set to the default value. The innovations algorithm is used to estimate white noise variance.

Value

Returns an ARMA model consisting of a list with the following components.

phi        0
theta      Vector of MA coefficients (index number equals coefficient subscript)
sigma2     White noise variance
aicc       Akaike information criterion corrected
se.phi     0
se.theta   Standard errors for the MA coefficients

See Also

arma burg hannan yw

Examples

M = c("diff",1)
e = Resid(dowj,M)
a = ia(e,1)
print(a)
lake

Description
Level of Lake Huron, 1875 to 1972

Examples
plotc(lake)

ma.inf

Description
Compute MA infinity coefficients

Usage
ma.inf(a, n = 50)

Arguments
a ARMA model
n Order

Details
The ARMA model is a list with the following components.

phi Vector of AR coefficients (index number equals coefficient subscript)
theta Vector of MA coefficients (index number equals coefficient subscript)
sigma2 White noise variance

Value
Returns a vector of length n+1 to accomodate coefficient 0 at index 1.

See Also
ar.inf
**Examples**

```r
def = c("diff", 12)
e = resid(deaths, def)
a = arma(e, 1, 1)
ma.inf(a, 10)
```

---

**Description**

Plot a periodogram

**Usage**

```r
periodogram(x, q = 0, opt = 2)
```

**Arguments**

- `x`: Time series data
- `q`: MA filter order
- `opt`: Plot option (0 silent, 1 periodogram only, 2 periodogram and filter)

**Details**

The filter `q` can be a vector in which case the overall filter is the composition of MA filters of the designated orders.

**Value**

The periodogram vector divided by 2pi is returned invisibly.

**See Also**

- `plots`

**Examples**

```r
periodogram(Sunspots, c(1, 1, 1))
```
**plota**

*Plot data and/or model ACF and PACF*

**Description**

Plot data and/or model ACF and PACF

**Usage**

```r
plota(u, v = NULL, h = 40)
```

**Arguments**

- `u, v` Data and/or ARMA model in either order
- `h` Maximum lag

**Value**

None

**Examples**

```r
plota(Sunspots)
a = yw(Sunspots, 2)
plota(Sunspots, a)
```

---

**plotc**

*Plot one or two time series*

**Description**

Plot one or two time series

**Usage**

```r
plotc(y1, y2 = NULL)
```

**Arguments**

- `y1` Data vector (plotted in blue with knots)
- `y2` Data vector (plotted in red, no knots)

**Value**

None
**Examples**

```r
plotc(uspop)
y = trend(uspop, 2)
plotc(uspop, y)
```

**Description**

Plot spectrum of data or ARMA model

**Usage**

```r
plots(u)
```

**Arguments**

- `u` Data vector or an ARMA model

**Value**

None

**See Also**

`periodogram`

**Examples**

```r
a = specify(ar=c(0, 0, .99))
plots(a)
```

---

**Resid**

**Description**

Compute residuals

**Usage**

```r
Resid(x, M = NULL, a = NULL)
```
Arguments

- x: Time series data
- M: Data model
- a: ARMA model

Details

The data model can be NULL for none. Otherwise M is a vector of function names and arguments.

Example:

```r
M = c("log", "season", 12, "trend", 1)
```

The above model takes the log of the data, then subtracts a seasonal component of period 12, then subtracts a linear trend component.

These are the available functions:

- **diff**: Difference the data. Has a single argument, the lag.
- **hr**: Subtract harmonic components. Has one or more arguments, each specifying the number of observations per harmonic.
- **log**: Take the log of the data, has no arguments.
- **season**: Subtract a seasonal component. Has a single argument, the number of observations per season.
- **trend**: Subtract a trend component. Has a single argument, the order of the trend (1 linear, 2 quadratic, etc.)

At the end of the model there is an implicit subtraction of the mean operation. Hence the resulting time series always has zero mean.

Value

Returns a vector of residuals the same length as x.

See Also

- test

Examples

```r
M = c("log", "season", 12, "trend", 1)
e = resid(wine, M)

a = arma(e, 1, 1)
e = resid(wine, M, a)
```

<table>
<thead>
<tr>
<th>season</th>
<th>Estimate seasonal component</th>
</tr>
</thead>
</table>

Description

Estimate seasonal component
Usage
season(x, d)

Arguments
x 
Time series data

d 
Number of observations per season

Value
Returns a vector the same length as x. Subtract from x to obtain residuals.

See Also
trend

Examples
y = season(deaths,12)
plotc(deaths,y)

Description
Run a self test

Usage
selftest()

Details
This function is a useful check if the code is modified.

Value
None

Examples
selftest()
**sim**  
*Generate synthetic observations*

**Description**
Generate synthetic observations

**Usage**

```
sim(a, n = 100)
```

**Arguments**
- `a` ARMA model
- `n` Number of synthetic observations required

**Details**
The ARMA model is a list with the following components.

- `phi` Vector of AR coefficients (index number equals coefficient subscript)
- `theta` Vector of MA coefficients (index number equals coefficient subscript)
- `sigma2` White noise variance

**Value**
Returns a vector of `n` synthetic observations.

**Examples**
```
a = specify(ar=c(0,0,.99))
x = sim(a,60)
plotc(x)
```

**smooth.exp**  
*Apply an exponential filter*

**Description**
Apply an exponential filter

**Usage**

```
smooth.exp(x, alpha)
```
Arguments

- **x**: Time series data
- **alpha**: Smoothness setting, 0-1

Details

Zero is maximum smoothness.

Value

Returns a vector of smoothed data the same length as `x`.

Examples

```r
y = smooth.exp(strikes, .4)
plotc(strikes, y)
```

---

**smooth.fft**

*Apply a low pass filter*

Description

Apply a low pass filter

Usage

```r
smooth.fft(x, f)
```

Arguments

- **x**: Time series data
- **f**: Cut-off frequency, 0-1

Details

The cut-off frequency is specified as a fraction. For example, `c=.25` passes the lowest 25% of the spectrum.

Value

Returns a vector the same length as `x`.

Examples

```r
y = smooth.fft(deaths, .1)
plotc(deaths, y)
```
Apply a moving average filter

**Description**

Apply a moving average filter

**Usage**

smooth.ma(x, q)

**Arguments**

- x: Time series data
- q: Filter order

**Details**

The averaging function uses 2q+1 values.

**Value**

Returns a vector the same length as x.

**Examples**

```r
y = smooth.ma(strikes, 2)
plotc(strikes, y)
```

---

Apply a spectral filter

**Description**

Apply a spectral filter

**Usage**

smooth.rank(x, k)

**Arguments**

- x: Time series data
- k: Number of frequencies
Details
Passes the mean and the k frequencies with the highest amplitude. The remainder of the spectrum is filtered out.

Value
Returns a vector the same length as x.

Examples
```r
y = smooth.rank(deaths,2)
plotc(deaths,y)
```

### specify

**Specify an ARMA model**

Description
Specify an ARMA model

Usage
```r
specify(ar = 0L, ma = 0, sigma2 = 1)
```

Arguments
- **ar** Vector of AR coefficients (index number equals coefficient subscript)
- **ma** Vector of MA coefficients (index number equals coefficient subscript)
- **sigma2** White noise variance

Value
Returns an ARMA model consisting of a list with the following components.
- **phi** Vector of AR coefficients (index number equals coefficient subscript)
- **theta** Vector of MA coefficients (index number equals coefficient subscript)
- **sigma2** White noise variance

Examples
```r
specify(ar=c(0,0,.99))
```
strikes  

USA union strikes, 1951-1980

Description

USA union strikes, 1951-1980

Examples

plotc(strikes)

Sunspots  

Number of sunspots, 1770 to 1869

Description

Number of sunspots, 1770 to 1869

Examples

plotc(Sunspots)

test  

Test residuals for stationarity and randomness

Description

Test residuals for stationarity and randomness

Usage

test(e)

Arguments

e  Time series data (typically residuals from Resid)

Details

Plots ACF, PACF, residuals, and QQ. Displays results for Ljung-Box, McLeod-Li, turning point, difference-sign, and rank tests. The plots can be used to check for stationarity and the other tests check for white noise.
trend

Value

None

See Also

Resid

Examples

```r
M = c("log","season",12,"trend",1)
e = Resid(wine,M)
test(e) ## Is e stationary?
a = arma(e,1,1)
ee = Resid(wine,M,a)
test(ee) ## Is ee white noise?
```

---

trend  Estimate trend component

Description

Estimate trend component

Usage

```r
trend(x, p)
```

Arguments

- `x` Time series data
- `p` Polynomial order (1 linear, 2 quadratic, etc.)

Value

Returns a vector the same length as `x`. Subtract from `x` to obtain residuals. The returned vector is the least squares fit of a polynomial to the data.

See Also

season

Examples

```r
y = trend(uspop,2)
plotc(uspop,y)
```
wine

Description
Australian red wine sales, January 1980 to October 1991

Examples
plotc(wine)

yw
Estimate AR coefficients using the Yule-Walker method

Description
Estimate AR coefficients using the Yule-Walker method

Usage
yw(x, p)

Arguments
x Time series data (typically residuals from Resid)
p AR order

Details
The innovations algorithm is used to estimate white noise variance.

Value
Returns an ARMA model consisting of a list with the following components.

phi Vector of AR coefficients (index number equals coefficient subscript)
theta 0
sigma2 White noise variance
aicc Akaike information criterion corrected
se.phi Standard errors for the AR coefficients
se.theta 0

See Also
arma burg hannan ia
Examples

```r
M = c("diff",1)
e = Resid(dowj,M)
a = yw(e,1)
```
### Index

+ Topic **datasets**
  - airpass, 5
  - deaths, 10
  - dowj, 11
  - lake, 15
  - strikes, 25
  - Sunspots, 25
  - wine, 27

+ Topic **package**
  - itsmr-package, 2

<table>
<thead>
<tr>
<th>Package</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>aacvf, 3</td>
<td></td>
</tr>
<tr>
<td>acvf, 4</td>
<td></td>
</tr>
<tr>
<td>airpass, 5</td>
<td></td>
</tr>
<tr>
<td>ar.inf, 5, 15</td>
<td></td>
</tr>
<tr>
<td>arar, 6</td>
<td></td>
</tr>
<tr>
<td>arma, 4, 7, 8, 9, 12–14, 27</td>
<td></td>
</tr>
<tr>
<td>autofit, 7, 8</td>
<td></td>
</tr>
<tr>
<td>burg, 7, 9, 13, 14, 27</td>
<td></td>
</tr>
<tr>
<td>check, 10</td>
<td></td>
</tr>
<tr>
<td>deaths, 10</td>
<td></td>
</tr>
<tr>
<td>dowj, 11</td>
<td></td>
</tr>
<tr>
<td>forecast, 6, 11</td>
<td></td>
</tr>
<tr>
<td>hannan, 7, 9, 12, 14, 27</td>
<td></td>
</tr>
<tr>
<td>hr, 13</td>
<td></td>
</tr>
<tr>
<td>ia, 7, 9, 13, 14, 27</td>
<td></td>
</tr>
<tr>
<td>itsmr (itsmr-package), 2</td>
<td></td>
</tr>
<tr>
<td>itsmr-package, 2</td>
<td></td>
</tr>
<tr>
<td>lake, 15</td>
<td></td>
</tr>
<tr>
<td>ma.inf, 5, 15</td>
<td></td>
</tr>
<tr>
<td>periodogram, 16, 18</td>
<td></td>
</tr>
<tr>
<td>plota, 4, 17</td>
<td></td>
</tr>
<tr>
<td>plotc, 17</td>
<td></td>
</tr>
<tr>
<td>plots, 16, 18</td>
<td></td>
</tr>
<tr>
<td>Resid, 12, 18, 26</td>
<td></td>
</tr>
<tr>
<td>season, 19, 26</td>
<td></td>
</tr>
<tr>
<td>selftest, 20</td>
<td></td>
</tr>
<tr>
<td>sim, 21</td>
<td></td>
</tr>
<tr>
<td>smooth.exp, 21</td>
<td></td>
</tr>
<tr>
<td>smooth.fft, 22</td>
<td></td>
</tr>
<tr>
<td>smooth.ma, 23</td>
<td></td>
</tr>
<tr>
<td>smooth.rank, 23</td>
<td></td>
</tr>
<tr>
<td>specify, 24</td>
<td></td>
</tr>
<tr>
<td>strikes, 25</td>
<td></td>
</tr>
<tr>
<td>Sunspots, 25</td>
<td></td>
</tr>
<tr>
<td>test, 12, 19, 25</td>
<td></td>
</tr>
<tr>
<td>trend, 20, 26</td>
<td></td>
</tr>
<tr>
<td>wine, 27</td>
<td></td>
</tr>
<tr>
<td>yw, 7, 9, 13, 14, 27</td>
<td></td>
</tr>
</tbody>
</table>